

A LEAF-SIZE HIERARCHY OF TWO-DIMENSIONAL ALTERNATING TURING MACHINES

Katsushi INOUE and Itsuo TAKANAMI

Department of Electronics, Faculty of Engineering, Yamaguchi University, Ube, 755 Japan

Juraj HROMKOVIČ

Department of Theoretical Cybernetics, Komenský University, 842 15 Bratislava, Czechoslovakia

Communicated by R. Siromoney

Received August 1987

Abstract. This paper introduces a simple, natural complexity measure for space bounded two-dimensional alternating Turing machines, called “leaf-size”, and provides a hierarchy of complexity classes based on leaf-size bounded computations. Specifically, we show that for any positive integer $k \geq 1$ and for any two functions $L: N \rightarrow N$ and $L': N \rightarrow N$ such that (1) L is a two-dimensionally space-constructible function such that $L(m)^{k+1} \leq m$ ($m \geq 1$), (2) $\lim_{m \rightarrow \infty} L(m)L'(m)^k / \log m = 0$ and (3) $\lim_{m \rightarrow \infty} L'(m)/L(m) = 0$, $L(m)$ space bounded and $L(m)^k$ leaf-size bounded two-dimensional alternating Turing machines are more powerful than $L(m)$ space bounded and $L'(m)^k$ leaf-size bounded two-dimensional alternating Turing machines.

1. Introduction

Alternating Turing machines were introduced [1] as a generalization of nondeterministic Turing machines and as a mechanism to model parallel computation. Further research into alternating machines have been continuing [1–3, 5–9, 11, 12]. However, there are many problems about alternating machines to be solved in the future.

In [6, 7, 3], we introduced a two-dimensional alternating Turing machine (2-ATM), and gave several properties of this machine. This paper continues the investigation of the fundamental properties of 2-ATMs whose input tapes are restricted to square ones. In particular, we shall introduce a simple, natural complexity measure for 2-ATMs, called “leaf-size”, and provide a hierarchy of complexity classes based on leaf-size bounded computations. Specifically, we show that for any positive integer $k \geq 1$ and for any two functions $L: N \rightarrow N$ and $L': N \rightarrow N$ such that (1) L is a two-dimensionally space constructible function such that $L(m)^{k+1} \leq m$ ($m \geq 1$), (2) $\lim_{m \rightarrow \infty} L(m)L'(m)^k / \log m = 0$, and (3) $\lim_{m \rightarrow \infty} L'(m)/L(m) = 0$, $L(m)$ space bounded and $L(m)^k$ leaf-size bounded two-dimensional alternating Turing machines are more powerful than $L(m)$ space bounded and $L'(m)^k$ leaf-size bounded two-dimensional alternating Turing machines.

Leaf-size is a useful abstraction which provides a spectrum of complexity classes intermediate between nondeterminism and full alternation. The concept of leaf-size bounded computations have already been introduced in [6, 5]. Similar concepts were introduced in [8, 2].

2. Preliminaries

Definition 2.1. Let Σ be a finite set of symbols. A *two-dimensional tape* over Σ is a two-dimensional rectangular array of elements of Σ .

The set of all two-dimensional tapes over Σ is denoted by $\Sigma^{(2)}$. Given a tape x in $\Sigma^{(2)}$, we let $l_1(x)$ be the number of rows of x and $l_2(x)$ be the number of columns of x . If $1 \leq i \leq l_1(x)$ and $1 \leq j \leq l_2(x)$, we let $x(i, j)$ denote the symbol in x with coordinates (i, j) . Further, we define $x[(i, j), (i', j')]$, only when $1 \leq i \leq i' \leq l_1(x)$ and $1 \leq j \leq j' \leq l_2(x)$, as the two-dimensional tape z satisfying the following:

- (i) $l_1(z) = i' - i + 1$ and $l_2(z) = j' - j + 1$;
- (ii) for each k, r ($1 \leq k \leq l_1(z)$, $1 \leq r \leq l_2(z)$),

$$z(k, r) = x(k + i - 1, r + j - 1).$$

We now recall a two-dimensional alternating Turing machine introduced in [6].

Definition 2.2. A *two-dimensional alternating Turing machine* (2-ATM) is a seven-tuple

$$M = (Q, q_0, U, F, \Sigma, \Gamma, \delta),$$

- (1) Q is a finite set of *states*,
- (2) $q_0 \in Q$ is the *initial state*,
- (3) $U \subseteq Q$ is the set of *universal states*,
- (4) $F \subseteq Q$ is the set of *accepting states*,
- (5) Σ is a finite *input alphabet* ($\# \notin \Sigma$ is the *boundary symbol*),
- (6) Γ is a finite *storage tape alphabet* ($B \in \Gamma$ is the *blank symbol*),
- (7) $\delta \subseteq (Q \times (\Sigma \cup \{\#\}) \times \Gamma) \times (Q \times (\Gamma - \{B\}) \times \{\text{left, right, up, down, no move}\} \times \{\text{left, right, no move}\})$ is the *next move relation*

A state q in $Q - U$ is said to be *existential*. As shown in Fig. 1, the machine M has a read-only (rectangular) input tape with boundary symbols “#” and one semi-infinite storage tape, initially blank. Of course, M has a finite control, an input head, and a storage tape head. A position is assigned to each cell of the read-only input tape and to each cell of the storage tape, as shown in Fig. 1. A *step* of M consists of reading one symbol from each tape, writing a symbol on the storage tape, moving the input and storage heads in specified directions, and entering a new state, in accordance with the next move relation δ . Note that the machine

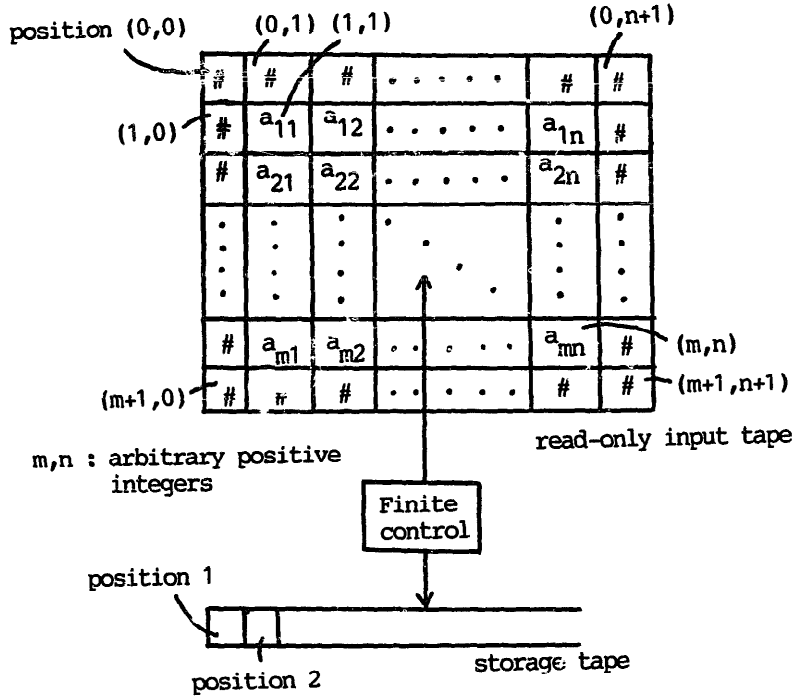


Fig. 1. Two-dimensional alternating Turing machine.

cannot write the blank symbol. If the input head falls off the input tape, or if the storage head falls off the storage tape (by moving left), then the machine can make no further move.

Definition 2.3. An *instantaneous description* (ID) of a 2-ATM $M = (Q, q_0, U, F, \Sigma, \Gamma, \delta)$ is an element of

$$\Sigma^{(2)} \times (N \cup \{0\})^2 \times S_M,$$

where $S_M = Q \times (\Gamma - \{B\})^* \times N$, and N denotes the set of all positive integers.

The first component of an ID $I = (x, (i, j), (q, \alpha, k))$ ¹ represents the input to M . The second component (i, j) of I represents the input head position. The third component (q, α, k) of I represents the state of the finite control, nonblank contents of the storage tape, and the storage-head position. An element of S_M is called a *storage state* of M . If q is the state associated with an ID I , then I is said to be a *universal (existential, accepting)* ID if q is a universal (existential, accepting) state. The *initial* ID of M on x is

$$I_M(x) = (x, (1, 1), (q_0, \lambda, 1)).$$

¹ We note that $0 \leq i \leq l_1(x) + 1$, $0 \leq j \leq l_2(x) + 1$ and $1 \leq k \leq |\alpha| + 1$, where for any string w , $|w|$ denotes the length of w (with $|\lambda| = 0$, where λ is the null string).

We write $I \vdash_M I'$ and say I' is a *successor* of I if an ID I' follows from an ID I in one step of M . A *computation tree* of M is a finite, nonempty labeled tree with the properties:

- (1) each node π of the tree is labeled with an ID $l(\pi)$;
- (2) if π is an internal node (a nonleaf) of the tree, $l(\pi)$ is universal and $\{I \mid l(\pi) \vdash_M I\} = \{I_1, \dots, I_k\}$, then π has exactly k children ρ_1, \dots, ρ_k such that $l(\rho_i) = I_i$;
- (3) if π is an internal node of the tree and $l(\pi)$ is existential, then π has exactly one child ρ such that $l(\pi) \vdash_M l(\rho)$.

A *computation tree of M on an input x* is a computation tree of M whose root is labeled with $I_M(x)$. An *accepting computation tree of M on x* is a computation tree of M on x whose leaves are all labeled with accepting IDs. We say that M *accepts x* if there is an accepting computation tree of M on input x . Define

$$T(M) = \{x \in \Sigma^{(2)} \mid M \text{ accepts } x\}.$$

In this paper, we are mainly concerned with a 2-ATM whose input tapes are restricted to square ones. We denote such a 2-ATM by 2-ATM^s.

Let $L: N \rightarrow N$ be a function with one variable m . With each 2-ATM^s M we associate a space complexity function SPACE which takes IDs to natural numbers. That is, for each ID $I = (x, (i, j), (q, \alpha, k))$, let $\text{SPACE}(I) = |\alpha|$. We say that M is $L(m)$ *space bounded* if for all m and for all x with $l_1(x) = l_2(x) = m$, if x is accepted by M , then there is an accepting computation tree of M on input x such that for each node π of the tree, $\text{SPACE}(l(\pi)) \leq L(m)$.

We next present a simple, natural complexity measure for 2-ATM^s, called leaf-size [6]. Leaf-size, in a sense, reflects the number of processors which run in parallel in reading a given input.

Definition 2.4. Let $Z: N \rightarrow N$ be a function with one variable m . For each finite tree t , let $\text{LEAF}(t)$ denote the leaf-size of t (i.e., the number of leaves of t). We say that a 2-ATM^s M is $Z(m)$ *leaf-size bounded* if, for each m and for each input x with $l_1(x) = l_2(x) = m$, each computation tree t of M on x is such that $\text{LEAF}(t) \leq Z(m)$.

By 2-ATM^s($L(m), Z(m)$) we denote a simultaneously $L(m)$ space bounded and $Z(m)$ leaf-size bounded 2-ATM^s. Define

$$\begin{aligned} \mathcal{L}[2\text{-ATM}^s(L(m), Z(m))] \\ = \{T \mid T = T(M) \text{ for some } 2\text{-ATM}^s(L(m), Z(m)) M\}. \end{aligned}$$

We need the following concepts in the next section.

Definition 2.5. A two-dimensional *deterministic* Turing machine [4] is a 2-ATM whose IDs each have at most one successor. A function $L: N \rightarrow N$ is *two-dimensionally space constructible* if there is a two-dimensional deterministic Turing machine M such that

(1) for each $m \geq 1$ and for each input tape x with $l_1(x) = l_2(x) = m$, M uses at most $L(m)$ cells of the storage tape,

(2) for each $m \geq 1$, there exists some input tape x with $l_1(x) = l_2(x) = m$ on which M halts after its read-write head has marked off exactly $L(m)$ cells of the storage tape, and

(3) for each $m \geq 1$, when given any input tape x with $l_1(x) = l_2(x) = m$, M never halts without marking off exactly $L(m)$ cells of the storage tape.

(In this case, we say that M constructs the function L .)

Definition 2.6. Let Σ_1, Σ_2 be finite sets of symbols. A *projection* is a mapping $\bar{\tau}: \Sigma_1^{(2)} \rightarrow \Sigma_2^{(2)}$ which is obtained by extending a mapping $\tau: \Sigma_1 \rightarrow \Sigma_2$ as follows:

$$\begin{aligned} \bar{\tau}(x) = x' \Leftrightarrow & \text{ (i) } l_k(x) = l_k(x') \text{ for each } k = 1, 2, \text{ and} \\ & \text{(ii) } \tau(x(i, j)) = x'(i, j) \text{ for each } (i, j) \\ & \in \{(i, j) \mid 1 \leq i \leq l_1(x) \text{ and } 1 \leq j \leq l_2(x)\}. \end{aligned}$$

3. Results

This section investigates a hierarchical property of the powers of space bounded 2-ATM's based on leaf-size bounded computations. Specifically, we show that $\mathcal{L}[2\text{-ATM}^s(L(m), L'(m)^k)] \subsetneq \mathcal{L}[2\text{-ATM}^s(L(m), L(m)^k)]$ for any positive integer $k \geq 1$ and for any two functions L and L' such that

- (i) L is a two-dimensionally space-constructible function such that $L(m)^{k+1} \leq m$ ($m \geq 1$),
- (ii) $\lim_{m \rightarrow \infty} L(m)L'(m)^k / \log m = 0$, and
- (iii) $\lim_{m \rightarrow \infty} L'(m)/L(m) = 0$.

We first give several preliminaries to obtain the desired result. Let Σ be a finite alphabet. For each $m \geq 2$ and each $1 \leq n \leq m-1$, an (m, n) -chunk over Σ is a pattern x over Σ as shown in Fig. 2, where $x_1 \in \Sigma^{(2)}$, $x_2 \in \Sigma^{(2)}$, $l_1(x_1) = m-1$, $l_2(x_1) = n$, $l_1(x_2) = m$ and $l_2(x_2) = m-n$. (Below, " (m, n) -chunk" means an (m, n) -chunk over Σ .) Let M be a $2\text{-ATM}^s(l, z)$. Note that if the numbers of states and storage-tape

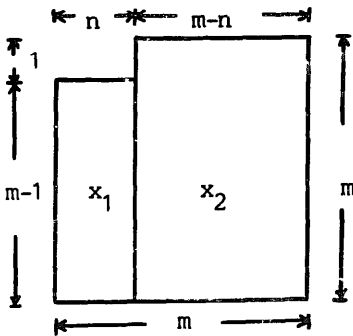


Fig. 2. (m, n) -chunk.

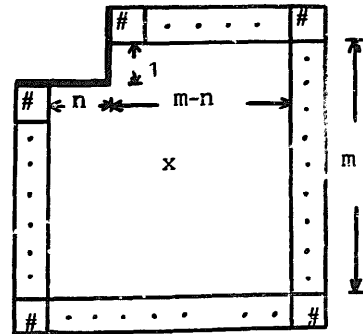
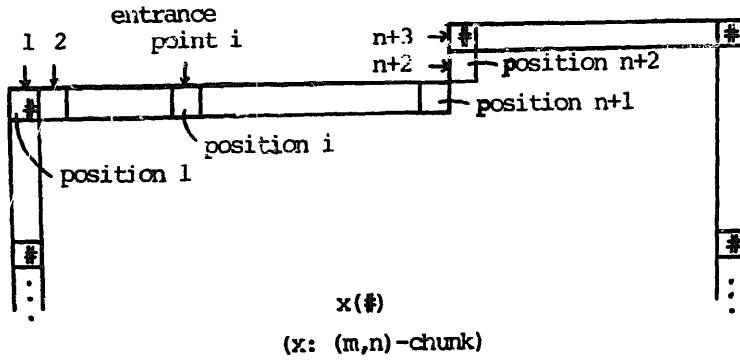


Fig. 3.

Fig. 4. Entrance points to $x(\#)$ and positioning of the cells of $x(\#)$.

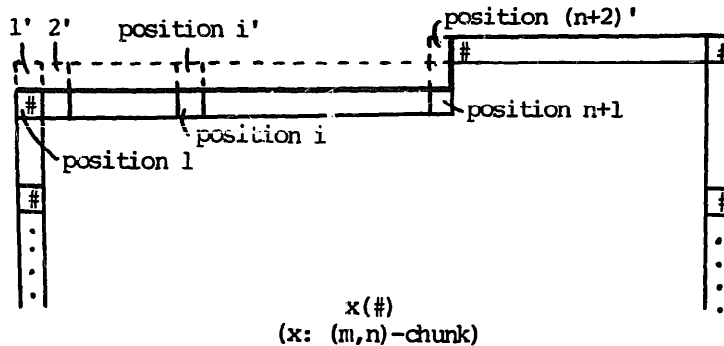
symbols of M are s and t , respectively, then the number of possible storage states of M is slt^l . Let Σ be the input alphabet of M , and let $\#$ be the boundary symbol of M . For any (m, n) -chunk x , we denote by $x(\#)$ the pattern (obtained from x by surrounding x with $\#$ s) as shown in Fig. 3. Below we assume without loss of generality that for any (m, n) -chunk ($m \geq 2$, $1 \leq n \leq m - 1$), M has the following property²:

- (A) M enters or exits the pattern $x(\#)$ only at the face designated by the bold line in Fig. 3, and M never enters an accepting state in $x(\#)$.

Then the number of entrance points to $x(\#)$ for M is $n+3$. We suppose that these entrance points are numbered $1, 2, \dots, n+3$ as shown in Fig. 4. For each (m, n) -chunk x , an ID of M on $x(\#)$ is of the form

$$(x(\#), (p, (q, \alpha, k))),$$

where p represents the position of the head of M on $x(\#)$, and (q, α, k) represents a storage state of M . The second component $(p, (q, \alpha, k))$ of an ID $I = (x(\#), (p, (q, \alpha, k)))$ is called the *configuration component* of I . For convenience sake, for each i ($1 \leq i \leq n+3$), let the position of the cell confronted with entrance point i of $x(\#)$ be " i " (see Fig. 4.) Further, as shown in Fig. 5, we consider $n+2$ virtual cells (confronted with $x(\#)$) designated by dotted line squares, and we assign

Fig. 5. Virtual cells of $x(\#)$ and positioning of virtual cells.

² Note that for any $2\text{-ATM}^s(l, z)$ M' , we can construct a $2\text{-ATM}^s(l, z)$ M with property (A) such that $T(M) = T(M')$.

positions $1', 2', \dots, n', (n+1)', (n+2)'$ to these virtual cells. We include these positions in the set of positions of the head of M on $x(\#)$.

An ID $I = (x(\#), (p, (q, \alpha, k)))$ is said to be *universal (existential)* if q is a universal (existential) state. For any two IDs I and I' of M on $x(\#)$, we write $I \vdash_M I'$ and say I' is a *successor* of I if I' follows from I in one step of M on $x(\#)$. Note that for any ID $I = (x(\#), (p, (q, \alpha, k)))$, where x is an (m, n) -chunk, such that $p \in \{1', 2', \dots, n', (n+1)', (n+2)'\}$ (i.e., p is a virtual position), I has no successor.

A *computation tree* of M on $x(\#)$ is a finite, nonempty labeled tree with the properties:

- (1) each node π of the tree is labeled with an ID, $l(\pi)$, of M on $x(\#)$;
- (2) if π is an internal node (a nonleaf) of the tree and $l(\pi)$ is universal and $\{I \mid l(\pi) \vdash_M I\} = \{I_1, I_2, \dots, I_k\}$, then π has exactly k children ρ_1, \dots, ρ_k such that $l(\rho_i) = I_i$;
- (3) if π is an internal node of the tree and $l(\pi)$ is existential, then π has exactly one child ρ such that $l(\pi) \vdash_M l(\rho)$.

A *prominent computation tree* of M on an (m, n) -chunk x is a computation tree of M on $x(\#)$ with the properties:

- (1) the root node is labeled with an ID of the form $(x(\#), (i, (q, \alpha, k)))$, where $1 \leq i \leq n+3$ (i.e., the root node is labeled with an ID of M just after M entered the pattern $x(\#)$ from some entrance point i);
- (2) each leaf node is labeled either
 - (a) with an ID of the form $(x(\#), (j, (q, \alpha, k)))$, where $j \in \{1', 2', \dots, (n+2)'\}$ (i.e., an ID of M just after M exited the pattern $x(\#)$), or
 - (b) with an ID I such that starting from the ID I , M never reaches a universal ID which has two or more successors, and M never exists $x(\#)$.

(A leaf node labeled with an ID of type (b) above is called a *looping leaf node*. A leaf node labeled with an ID of type (a) above is called a *normal leaf node*.)

Let $C = \{c_1, c_2, \dots, c_u\}$ be the set of possible storage states of M , where $u = slt^1$. For each prominent computation tree t of M on an (m, n) -chunk, let the *leaf configuration set* of t (denoted by $LCS(t)$) be a “multiset” of elements of $\{1', 2', \dots, (n+2)'\} \times C \cup \{L\}$ (where L is a new symbol) defined as follows:

- (1) for each normal leaf node π of t , $LCS(t)$ contains the configuration component of $l(\pi)$;
- (2) for each looping leaf node of t , $LCS(t)$ contains the symbol L ;
- (3) $LCS(t)$ does not contain any element other than elements described in (1) and (2) above.

(Note that for any prominent computation tree t of M , $|LCS(t)| \leq z$, since M is z leaf-size bounded.)³

For each (m, n) -chunk x and for each $(i, c) \in \{1, 2, \dots, n+3\} \times C$, let

$$M_{(i,c)}(x) = \{LCS(t) \mid t \text{ is a prominent computation tree of } M \text{ on } x \text{ whose root is labeled with the ID } (x(\#), (i, c))\}.$$

³ For any set S , $|S|$ denotes the number of elements of S .

Let x, y be two (m, n) -chunks. We say that x and y are M -equivalent if for each $(i, c) \in \{1, 2, \dots, n+3\} \times C$, $M_{(i,c)}(x) = M_{(i,c)}(y)$.

For any (m, n) -chunk x and for any tape $v \in \Sigma^{(2)}$ with $l_1(v) = 1$ and $l_2(v) = n$, let $x[v]$ be the tape in $\Sigma^{(2)}$ consisting of v and x as shown in Fig. 6.

The following lemma means that M cannot distinguish between two (m, n) -chunks which are M -equivalent.

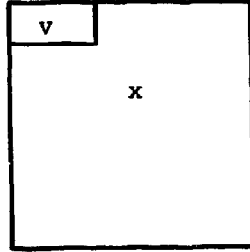


Fig. 6. $x[v]$.

Lemma 3.1. *Let M be a $2\text{-ATM}^s(l, z)$ with the property (A) described before, and Σ be the input alphabet of M . Let x and y be M -equivalent (m, n) -chunks over Σ ($m \geq 2, 1 \leq n \leq m-1$). Then, for any tape $v \in \Sigma^{(2)}$ with $l_1(v) = 1$ and $l_2(v) = n$, $x[v]$ is accepted by M if and only if $y[v]$ is accepted by M .*

Proof. (*If part*). We assume that $y[v]$ is accepted by M . Then there exists an accepting computation tree t of M on $y[v]$ such that $\text{LEAF}(t)$ (i.e., the number of leaves of t) $\leq z$. Since x and y are M -equivalent, we can construct from t an accepting computation tree t' of M on $x[v]$ such that $\text{LEAF}(t') = \text{LEAF}(t) \leq z$. Therefore, $x[v]$ is accepted by M .

(*Only-if part*). Analogous to “if part”. \square

Clearly, M -equivalence is an equivalence relation on (m, n) -chunks, and we obtain the following lemma.

Lemma 3.2. *Let M be a $2\text{-ATM}^s(l, z)$ with the property (A) described before, and Σ be the input alphabet of M . Further, let s and t be the numbers of states and storage tape symbols of M , respectively, and let $u = slt^l$. Then there are at most $(2^{b^{z+1}})^d$ M -equivalence classes of (m, n) -chunks over Σ , where $b = (n+2)u+1$ and $d = (n+3)u$.*

Proof. The lemma follows from the observation that

(1) $|\{1, 2, \dots, n+3\} \times C| = (n+3)u = d$ (where C is the set of possible storage states of M), and

(2) the number of possible leaf configuration sets of prominent computation trees of M on (m, n) -chunks is bounded by

$$b + b^2 + \dots + b^z \leq b^{z+1} \quad (\text{where } b = (n+2)u+1)$$

since M is z leaf-size bounded. \square

We are now ready to prove the main theorem.

Theorem 3.3. *Let $k \geq 1$ be a positive integer. Let $L: N \rightarrow N$ and $L': N \rightarrow N$ be any functions such that*

- (1) *L is a two-dimensionally space-constructible function such that $L(m)^{k+1} \leq m$ ($m \geq 1$),*
- (2) *$\lim_{m \rightarrow \infty} L(m)L'(m)^k / \log m = 0$, and*
- (3) *$\lim_{m \rightarrow \infty} L'(m) / L(m) = 0$.*

Then there is a set in $\mathcal{L}[2\text{-ATM}^s(L(m), L(m)^k)]$, but not in $\mathcal{L}[2\text{-ATM}^s(L(m), L'(m)^k)]$.

Proof. Let M be a two-dimensional deterministic Turing machine which constructs the function L . Let $T_k[L, M]$ be the following set, which depends on k , L and M :

$$T_k[L, M] = \{x \in (\Sigma \times \{0, 1\})^{(2)} \mid \exists m \geq 2 [l_1(x) = l_2(x) = m \text{ \& (when the tape } \bar{h}_1(x) \text{ is presented to } M, \text{ its read-write head marks off exactly } L(m) \text{ cells of the storage tape and then halts)} \\ \& \exists i (2 \leq i \leq m) [\bar{h}_2(x[(1, 1), (1, L(m)^{k+1})]) = \bar{h}_2(x[(i, 1), (i, L(m)^{k+1})])]\}],$$

where Σ is the input alphabet of M , and $\bar{h}_1(\bar{h}_2)$ is the projection which is obtained by extending the mapping $h_1: \Sigma \times \{0, 1\} \rightarrow \Sigma$ ($h_2: \Sigma \times \{0, 1\} \rightarrow \{0, 1\}$) such that for any $c = (a, b) \in \Sigma \times \{0, 1\}$, $h_1(c) = a$, ($h_2(c) = b$). Below, we shall show that $T_k[L, M] \in \mathcal{L}[2\text{-ATM}^s(L(m), L(m)^k)]$ and $T_k[L, M] \notin \mathcal{L}[2\text{-ATM}^s(L(m), L'(m)^k)]$.

The set $T_k[L, M]$ is accepted by a $2\text{-ATM}^s(L(m), L(m)^k)$ M_1 which acts as follows. Suppose that an input x with $l_1(x) = l_2(x) = m$ ($m \geq 2$) is presented to M_1 . M_1 directly simulates the action of M on $\bar{h}_1(x)$. If M does not halt, then M_1 also does not halt, and will not accept x . If M_1 finds out that M halts (in this case, note that M_1 has marked off exactly $L(m)$ cells of the storage tape because M constructs the function L), then M_1 existentially chooses some i ($2 \leq i \leq m$) and moves its input tape head on the first column of the i th row of x . After that, M_1 universally tries to check that, for each $1 \leq j \leq L(m)^k$,

$$\bar{h}_2(x[(i, (j-1)L(m)+1), (i, jL(m))]) = \bar{h}_2(x[(1, (j-1)L(m)+1), (1, jL(m))]).$$

That is, on the i th row and $((j-1)L(m)+1)$ st column of x ($1 \leq j \leq L(m)^k$), M_1 enters a universal state to choose one of two further actions. One action is to pick up and store the segment

$$\bar{h}_2(x[(i, (j-1)L(m)+1), (i, jL(m))])$$

on some track of the storage tape (of course, M_1 uses exactly $L(m)$ cells marked off), to compare the segment stored above with the segment

$$\bar{h}_2(x[(1, (j-1)L(m)+1), (1, jL(m))]),$$

and to enter an accepting state only if both segments are identical. The other action is to continue moving to the i th row and $(jL(m)+1)$ st column of x (in order to pick up the next segment

$$\bar{h}_2(x[(i, jL(m)+1), (i, (j+1)L(m))])$$

and compare it with the corresponding segment

$$\bar{h}_2(x[(1, jL(m)+1), (1, (j+1)L(m))])).$$

Note that the number of pairs of segments which should be compared with each other in the future can be easily seen by using $L(m)$ cells of the storage tape. It will be obvious that the input x is in $T_k[L, M]$ if and only if there is an accepting computation tree of M_1 on x with $L(m)^k$ leaves. Thus $T_k[L, M] \in \mathcal{L}[2\text{-ATM}^s(L(m), L(m)^k)]$.

We next show that $T_k[L, M] \notin \mathcal{L}[2\text{-ATM}^s(L(m), L'(m)^k)]$. Suppose that there is a $2\text{-ATM}^s(L(m), L'(m)^k)$ M_2 accepting $T_k[L, M]$. Let s and t be the numbers of states (of the finite control) and storage tape symbols of M_2 , respectively. We assume without loss of generality that when M_2 accepts a tape x in $T_k[L, M]$, it enters an accepting state only on the upper left-hand corner of x , and that M_2 never falls off an input tape out of the boundary symbol $\#$. (Thus M_2 satisfies the property (A) described before.) For each $m \geq 2$, let $w(m) \in \Sigma^{(2)}$ be a fixed tape such that

- (i) $l_1(w(m)) = l_2(w(m)) = m$ and
- (ii) when $w(m)$ is presented to M , it marks off exactly $L(m)$ cells of the storage tape and halts.

(Note that for each $m \geq 2$, there exists such a tape $w(m)$ because M constructs the function L .) For each $m \geq 2$, let

$$\begin{aligned} V(m) &= \{x \in (\Sigma \times \{0, 1\})^{(2)} \mid l_1(x) = l_2(x) = m \\ &\quad \& \bar{h}_2(x[(1, 1), (m, L(m)^{k+1})]) \in \{0, 1\}^{(2)} \\ &\quad \& \bar{h}_2(x[(1, L(m)^{k+1}+1), (m, m)]) \in \{0\}^{(2)} \& \bar{h}_1(x) = w(m)\}^4, \\ Y(m) &= \{y \in \{0, 1\}^{(2)} \mid l_1(y) = 1 \& l_2(y) = L(m)^{k+1}\}, \\ R(m) &= \{\text{row}(x) \mid x \in V(m)\}, \end{aligned}$$

where for each x in $V(m)$,

$$\text{row}(x) = \{y \in Y(m) \mid y = \bar{h}_2(x[(i, 1), (i, L(m)^{k+1})]) \text{ for some } i \ (2 \leq i \leq m)\}.$$

⁴ By the assumption that $L(m)^{k+1} \leq m$ ($m \geq 1$), $V(m)$ is well defined.

Since $|Y(m)| = 2^{L(m)^{k+1}}$, it follows that

$$|R(m)| = \begin{cases} \binom{2^{L(m)^{k+1}}}{1} + \binom{2^{L(m)^{k+1}}}{2} + \cdots + \binom{2^{L(m)^{k+1}}}{m-1}, & \text{if } 2^{L(m)^{k+1}} > m-1, \\ \binom{2^{L(m)^{k+1}}}{1} + \cdots + \binom{2^{L(m)^{k+1}}}{2^{L(m)^{k+1}}} = 2^{2^{L(m)^{k+1}}} - 1, & \text{otherwise.} \end{cases}$$

Note that $B = \{p \mid \text{for some } x \text{ in } V(m), p \text{ is the pattern obtained from } x \text{ by cutting the segment } x[(1, 1), (1, L(m)^{k+1})] \text{ off}\}$ is a set of $(m, L(m)^{k+1})$ -chunks over $\Sigma \times \{0, 1\}$. Since M_2 can use at most $L(m)$ cells of the storage tape and M_2 is $L'(m)^k$ lead-size bounded when M_2 reads a tape in $V(m)$, from Lemma 3.2, there are at most

$$E(m) = (2^{b[m]^{L'(m)^{k+1}}})^{d[m]}$$

M_2 -equivalence classes of $(m, L(m)^{k+1})$ -chunks (over $\Sigma \times \{0, 1\}$) in B , where $b[m] = (L(m)^{k+1} + 2)u[m] + 1$, $d[m] = (L(m)^{k+1} + 3)u[m]$ and $u[m] = sL(m)t^{L(m)}$. We denote these M_2 -equivalence classes by $C_1, C_2, \dots, C_{E(m)}$. Since $\lim_{m \rightarrow \infty} L(m)L'(m)^k / \log m = 0$ and $\lim_{m \rightarrow \infty} L'(m)/L(m) = 0$ (by assumption), it follows that for large m , $|R(m)| > E(m)$. For such m , there must be some Q, Q' ($Q \neq Q'$) in $R(m)$ and some C_i ($1 \leq i \leq E(m)$) such that the following statement holds: There exist two tapes x, y in $V(m)$ such that

- (i) $x[(1, 1), (1, L(m)^{k+1})] = y[(1, 1), (1, L(m)^{k+1})]$ and $\bar{h}_2(x[(1, 1), (1, L(m)^{k+1})]) = \bar{h}_2(y[(1, 1), (1, L(m)^{k+1})]) = \rho$ for some ρ in Q but not in Q' ,
- (ii) $\text{row}(x) = Q$ and $\text{row}(y) = Q'$, and
- (iii) both p_x and p_y are in C_i , where $p_x(p_y)$ is the $(m, L(m)^{k+1})$ -chunk over $\Sigma \times \{0, 1\}$ obtained from x (from y) by cutting the segment $x[(1, 1), (1, L(m)^{k+1})]$ (the segment $y[(1, 1), (1, L(m)^{k+1})]$) off.

As is easily seen, x is in $T_k[L, M]$, and so x is accepted by M_2 . Therefore, from Lemma 3.1, it follows that y is also accepted by M_2 , which is a contradiction. (Note that y is not in $T_k[L, M]$.) Thus $T_k[L, M] \notin \mathcal{L}[2\text{-ATM}^s(L(m), L'(m)^k)]$. This completes the proof of the theorem. \square

Corollary 3.4. *Let $k \geq 1$ be a positive integer. Let $L: N \rightarrow N$ and $L': N \rightarrow N$ be any functions satisfying the condition that $L'(m) \leq L(m)$ ($m \geq 1$) and satisfying conditions (1), (2) and (3) described in Theorem 3.3. Then*

$$\mathcal{L}[2\text{-ATM}^s(L(m), L'(m)^k)] \subsetneq \mathcal{L}[2\text{-ATM}^s(L(m), L(m)^k)].$$

For each r in N , let $\log^{(r)}m$ be the function defined as follows:

$$\log^{(1)}m = \begin{cases} 0 & m = 0, \\ \lceil \log m \rceil, & m \geq 1, \end{cases}$$

$$\log^{(r+1)}m = \log^{(1)}(\log^{(r)}m)$$

where $\lceil \log m \rceil$ denotes the smallest integer greater than or equal to $\log m$. As shown in [10, Theorem 3], the function $\log^{(r)} m$ ($r \geq 1$) is two-dimensionally space-constructible. It is easy to see that for each $r \geq 1$, $\log^{(r+1)} m \leq \log^{(r)} m$ ($m \geq 1$) and $\lim_{m \rightarrow \infty} \log^{(r+1)} m / \log^{(r)} m = 0$. Further, for each $r \geq 2$ and each $k \geq 1$, $\lim_{m \rightarrow \infty} \log^{(r)} m (\log^{(r+1)} m)^k / \log m = 0$. From these facts and Corollary 3.4, we have the following.

Corollary 3.5. *For any $r \geq 2$ and any $k \geq 1$,*

$$\mathcal{L}[2\text{-ATM}^s(\log^{(r)} m, (\log^{(r+1)} m)^k)] \subsetneq \mathcal{L}[2\text{-ATM}^s(\log^{(r)} m, (\log^{(r)} m)^k)].$$

It is unknown whether a result analogous to Corollary 3.5 also holds for $r = 1$ and $k \geq 1$. It will also be interesting to investigate leaf-size hierarchy properties of the classes of sets accepted by 2-ATM^s with spaces of size greater than $\log m$.

References

- [1] A.K. Chandra, D.C. Kozen and L.J. Stockmeyer, Alternation, *J. Assoc. Comput. Mach.* **28**(1) (1981) 114–133.
- [2] J. Hromkovič, On the power of alternation in automata theory, *J. Comput. Systems Sci.* **31**(1) (1985) 28–39.
- [3] K. Inoue, A. Ito, I. Takanami and H. Taniguchi, A space-hierarchy result on two-dimensional alternating Turing machines with only universal states, *Inform. Sci.* **35** (1985) 79–90.
- [4] K. Inoue and I. Takanami, Three-way tape-bounded two-dimensional Turing machines, *Inform. Sci.* **17** (1979) 195–220.
- [5] K. Inoue, I. Takanami and H. Taniguchi, A note on alternating on-line Turing machines, *Inform. Process. Lett.* **15**(4) (1982) 164–168.
- [6] K. Inoue, I. Takanami and H. Taniguchi, Two-dimensional alternating Turing machines, *Theoret. Comput. Sci.* **27** (1983) 61–83.
- [7] A. Ito, K. Inoue, I. Takanami and H. Taniguchi, Two-dimensional alternating Turing machines with only universal states, *Inform. and Control* **55**(1–3) (1982) 193–221.
- [8] K.N. King, Measures of parallelism in alternating computation trees, in: *Proc. 13th Ann. ACM Symp. on Theory of Computing* (1981) 189–201.
- [9] R.E. Ladner, R.J. Lipton and L.J. Stockmeyer, Alternating pushdown automata, in: *Proc. 19th IEEE Symp. on Foundations of Computer Science*, Ann Arbor, MI (1978).
- [10] K. Morita, H. Umeo, H. Ebi and K. Sugata, Lower bounds on tape complexity of two-dimensional tape Turing machines, *IECE Japan Trans. D* (1978) 381.
- [11] W. Paul and R. Reischuk, On alternation, Part I, *Acta Inform.* **14** (1980) 243–255.
- [12] W.L. Ruzzo, Tree-size bounded alternation, *J. Comput. Systems Sci.* **21** (1980) 218–235.